Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 378 (2007) 198–207
Theoretical Computer Science
www.elsevier.com/locate/tcs

An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices☆

Robert W. Zhu, Guomin Yang, Duncan S. Wong*

Department of Computer Science, City University of Hong Kong, Hong Kong, China

Abstract

For an ID-based key exchange (KE) protocol, KGS forward secrecy is about the protection of previously established session keys after the master secret key of the Key Generation Server (KGS) is compromised. This is the strongest notion of forward secrecy that one can provide for an ID-based KE protocol. Among all the comparable protocols, there are only a few of them that provide this level of forward secrecy, and all of these protocols require expensive bilinear pairing operations and map-to-point hash operations that may not be suitable for implementation on low-power devices such as sensors. In this paper, we propose a new ID-based KE protocol which does not need any pairing or map-to-point hash operations. It also supports the strongest KGS forward secrecy. On its performance, we show that it is faster than previously proposed protocols in this category. Our protocol is a signature-based one, in which the signature scheme is a variant of a scheme proposed by Bellare et al. in Eurocrypt 2004. We show that the variant we proposed is secure, and also requires either less storage space or runtime computation than the original scheme.

1. Introduction

Since the first set of identity-based (ID-based) Key Exchange (KE) protocols were proposed [24,20,18,17] in the late '80s and early '90s, there has been a revival of interest in ID-based KE protocols recently [27,28,15,23] due to the discovery of several new applications of pairings on elliptic curves [10].

On security, most of them [24,18,17,27,28,23][1] only support *partial* or *perfect* forward secrecy but not **KGS forward secrecy** [20,15], the strongest notion of forward secrecy in the context of ID-based KE protocols. By partial forward secrecy, the previously established session keys will remain secure after the secret key of one communicating party is compromised. By perfect forward secrecy, the previously established session keys will remain secure after the secret keys of both communicating parties are compromised. By KGS (Key Generation Server) forward secrecy,

☆ A preliminary version appears in the Proc. of the First Workshop on Internet and Network Economics (WINE 2005) [R.W. Zhu, G. Yang, D.S. Wong, An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices, in: Internet and Network Economics: First International Workshop, WINE 2005, in: LNCS, vol. 3828, Springer-Verlag, 2005, pp. 500–509].

* Corresponding author. Tel.: +852 2788 8020; fax: +852 2788 8614.
E-mail addresses: zhuwei@cs.cityu.edu.hk (R.W. Zhu), csyanggm@cs.cityu.edu.hk (G. Yang), duncan@cs.cityu.edu.hk, duncan@cityu.edu.hk (D.S. Wong).

[1] Some protocols in [24,18] may be able to provide KGS forward secrecy under some condition specified in [12].

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.02.021

the previously established session keys will still remain secure even after the long-term secret key of the KGS is compromised. Note that compromising the KGS secret key implies compromising the secret keys of all parties in an ID-based cryptosystem. Hence, KGS forward secrecy is the strongest notion among these three.

On performance, according to the state-of-the-art results in [1,2], one bilinear pairing operation requires at least 10 times more multiplications in the underlying finite field than an elliptic curve point scalar multiplication does in the same finite field. For low-power devices such as sensors, cellphones, and low-end PDAs, which are usually characterized by limited battery lifetime and low computational power, applications using bilinear pairings can be too expensive to implement. In addition, most of the ID-based cryptosystems require a special hash function called map-to-point hash function [10,11] to convert a user's identifying information to a point on the underlying elliptic curve. This operation is also time consuming and cannot be treated as a conventional hash operation, which is commonly ignored in performance evaluation. A map-to-point hash function, on the other hand, is usually implemented as a probabilistic algorithm and is more expensive than a point scalar multiplication in terms of computation time. For example, in Smart's protocol [27], each communicating party needs to carry out two pairing operations, one scalar multiplication and one map-to-point hash operation in each protocol run. However, in many related protocols, the performance of Smart's protocol was evaluated by counting only the pairings and scalar multiplications while having all the map-to-point hash operations ignored. This is partly because all the pairing-based KE protocols require to carry out map-to-point hash operations.

If we focus on signature-based KE protocols, we can see that the efficiency of the underlying signature schemes has a significant impact on the overall efficiency of the KE protocols. As one bilinear pairing operation costs much more than one elliptic curve point scalar multiplication does, it may also be undesirable to adopt a bilinear pairing based signature scheme [3] to construct a KE protocol if the more efficient signature scheme is available. We will see later in this paper that in our ID-based KE protocol, we use a signature scheme which does not require any bilinear pairing operation.

Our contributions. We propose a new ID-based KE protocol which does not require any pairing or map-to-point hash. It also supports the strongest KGS forward secrecy. The protocol is also shown to be secure under the model defined by Canetti and Krawzcyk [14]. Our protocol is signature-based, in which the signature scheme is ID-based and is a variant of the BNN-IBS proposed by Bellare et al. in [5]. We show that our scheme is secure and also requires either less storage space or runtime computation than the BNN-IBS. On the performance of our ID-based KE protocol, we show that it is faster than all other comparable protocols.

Paper organization. In Section 2, we review some previously proposed ID-based KE protocols. In Section 3, we give a definition for ID-based signature schemes. This is followed by the description of our ID-based signature scheme and its security analysis in Section 4. In Section 5, we propose an ID-based KE protocol and analyze its security using a modular approach proposed by Bellare et al. [4]. Performance evaluation of the protocol is given in Section 6. We conclude the paper in Section 7.

2. Related work

The concept of ID-based cryptography was introduced by Shamir in 1984 [26]. The idea is to let a user's identity be his public key, and have his corresponding private key be generated by a publicly trusted Key Generation Server (KGS) using the user's identity and the master secret key of the KGS. An ID-based KE protocol [24,27,15] allows two communicating parties to generate a random session key such that the key is only known to them. On security, we generally require that such a protocol should be secure against both passive and active adversaries under some multi-session setting [7,8,4,14].

In 2002, Smart [27] proposed an ID-based KE protocol based on pairings. It only supports partial forward secrecy. In the same year, Yi [28] proposed a modification on Smart's protocol. The modified protocol is more efficient and supports perfect forward secrecy. Recently, a proven secure ID-based KE protocol was proposed [23], which also supports perfect forward secrecy. However, none of these schemes is able to support the strongest KGS forward secrecy.

In [15], Chen and Kudla proposed a protocol which supports KGS forward secrecy and has also been proven secure under a model similar to the Bellare–Rogaway model [7,8]. On performance, their protocol requires each

communicating party to carry out one pairing operation, four scalar multiplications and one map-to-point hash operation.

As a remark, in any of the protocols reviewed above, each communicating party always needs to carry out at least one pairing, one scalar multiplication, and one map-to-point hash. As mentioned in the beginning of this section, one pairing operation is at least 10 times more expensive than one scalar multiplication; and one map-to-point hash operation is also more expensive than one scalar multiplication. Hence, most of them do not have much advantage in terms of performance when compared with the old, non-pairing based KE protocols [24,20,18,17].

For protocols proposed in [24,18,17], expensive modular exponentiations are carried out by each of the communicating parties. Hence, they may not be suitable for implementation on low-power devices either. Protocols proposed in [20], on the other hand, can be implemented under an elliptic curve group and one of the protocols is believed to support the KGS forward secrecy. However, the performance of that protocol is slightly less efficient than our protocol. In addition, it is not known to be provably secure.

Comparing all these protocols with our protocol described in Section 5, we will see that our protocol not only supports the KGS forward secrecy, but is also faster than all the comparable protocols mentioned above. We will also show its security under the model defined by Canetti and Krawczyk [14].

We skip the review of related work on ID-based signature schemes and only go through their security definitions in the next section. Readers may refer to [5,3] for more information. In 4, we propose an ID-based scheme which does not need any pairing or map-to-point hash operation, which is known to be very costly. It can be seen that our ID-based scheme compares favourably even with those recently proposed schemes based on pairings, for example, [3] and ID-based variants of short signatures based on [9,29], that may be constructed, for example, from the certificate based approach described in [5].

3. IBS: Security model

To be self-contained, we review the definition and security model of an identity-based signature scheme given by Bellare et al. [5] in this section. For readers who are already familiar with these definitions, they can safely skip this section.

An *ID-based signature (IBS) scheme* is a tuple $(MKGen, UKGen, Sig, Ver)$ of polynomial-time algorithms. The first three may be randomized but the last one is not.

- A Key Generation Server (KGS) runs the master-key generation algorithm $MKGen$ on input 1^k, where $k \in \mathbb{N}$ is a security parameter, to obtain a master public/secret key pair (mpk, msk).
- The KGS can then run the user-key generation algorithm $UKGen$ on msk and identity $ID \in \{0,1\}^*$ to generate a secret key usk for a user identified by ID. It is assumed that usk is securely transported to that user.
- On input usk and message $m \in \{0,1\}^*$, the signing algorithm Sig returns a signature σ.
- On input mpk, ID, m and σ, the verification algorithm Ver returns an accept/reject decision to indicate whether a signature σ is valid for identity ID and message m.

For correctness, we require that for all $k \in \mathbb{N}$, $m \in \{0,1\}^*$, $ID \in \{0,1\}^*$, if $(mpk, msg) \leftarrow MKGen(1^k)$, $usk \leftarrow UKGen(msk, ID)$ and $\sigma \leftarrow Sig(usk, m)$, then $Ver(mpk, ID, m, \sigma) = 1$. We now provide the formal definition of a secure IBS scheme in terms of existential unforgeability against the chosen message and ID attacks (in short euf-cma-ida).

Definition 1. Let $(MKGen, UKGen, Sig, Ver)$ be an IBS scheme, $\mathcal{A}$ an adversary, and $k \in \mathbb{N}$ a security parameter. Consider the game below which is run by a simulator/challenger $\mathcal{S}$.

- $\mathcal{S}$ executes $MKGen(1^k)$ to get the master public/secret key pair (mpk, msk).
- $\mathcal{S}$ runs $\mathcal{A}$ on 1^k and mpk. During the simulation, $\mathcal{A}$ can make queries to the following oracles.
  - CreateUser(ID): If ID is not yet created, $\mathcal{S}$ executes $UKGen(msk, ID)$ to get the user's secret key usk_{ID}. ID is said to be created from now on and 1 is returned. Otherwise, (that is, if ID has already been created) 0 is returned.
  - Corrupt(ID): If ID has been created, usk_{ID} is returned and ID is said to be corrupted; otherwise, $\perp$ is returned for failure.

- Sign(ID, m): If ID has not been created, return $\perp$ for failure. Otherwise, $\mathcal{S}$ executes $Sig(usk_{ID}, m)$ to get a signature σ and returns σ. Then, m is said to be signed by ID.

– $\mathcal{A}$ is to output a triple (ID^*, m^*, σ^*).

$\mathcal{A}$ wins if $Ver(mpk, ID^*, m^*, \sigma^*) = 1$, ID^* is created but *not* corrupted and m^* is *not* signed by ID^*. The IBS scheme is euf-cma-ida *secure* if, for all probabilistic polynomial-time (PPT) algorithms $\mathcal{A}$, it is negligible for $\mathcal{A}$ to win the game.

In the next section, we describe a new IBS scheme which is euf-cma-ida secure, as defined in this section.

4. An IBS scheme based on EC-DLP

In the following, we first define some notations and then describe the four algorithms (*MKGen*, *UKGen*, *Sig*, *Ver*) of our IBS scheme.

Preliminaries. Let $k \in \mathbb{N}$ be a security parameter, $ID \in \{0,1\}^*$ an identity, and $m \in \{0,1\}^*$ a message. Let $\mathbf{F}$ be a finite field, $\mathscr{C}$ an elliptic curve defined over $\mathbf{F}$, and P an element of a large prime order p in $\mathscr{C}$. Let G be a cyclic subgroup of $\mathscr{C}$ generated by the 'base' point P, such that the elliptic curve discrete log problem (EC-DLP) is intractable. We assume that $(\mathscr{C}, \mathbf{F}, P, p)$ is a sequence of system-wide parameters[2]. Let $H_1 : \{0,1\}^* \to Z_p$ and $H_2 : \{0,1\}^* \to Z_p$ be two hash functions. For security analysis, we view them as random oracles [6].

Master-key generation
$(mpk, msk) \leftarrow MKGen(1^k)$: The trusted KGS (Key Generation Server) randomly picks $x \in_R \mathbb{Z}_p$ and computes $P_{\text{pub}} = xP$. The master public key mpk is set to P_{pub}, and the master secret key msk is set to (x, P_{pub}).

User-key generation
$usk \leftarrow UKGen(msk, ID)$: If ID is not created, the KGS sets the user's secret key usk to $(c, s, ID, P_{\text{pub}})$, where $c = H_1(P_{\text{pub}}, ID, cP_{\text{pub}} + sP)$ and $s \in \mathbb{Z}_p$. This secret key is generated as follows.

1. Randomly pick $r \in_R \mathbb{Z}_p$, compute $R = rP$ and $c \leftarrow H_1(P_{\text{pub}}, ID, R)$.
2. Compute $s = r - cx \mod p$.

Signature generation
$\sigma \leftarrow Sig(usk, m)$: Given the user's secret key $usk = (c, s, ID, P_{\text{pub}})$, a signature $\sigma = (c, T, \pi)$ on message m is generated as follows.

1. Randomly pick $t \in_R \mathbb{Z}_p$, compute $T = tP$.
2. Compute $e \leftarrow H_2(P_{\text{pub}}, ID, m, T, c)$ and $\pi = t - es \mod p$.

Signature verification
$1/0 \leftarrow Ver(mpk, ID, m, \sigma)$: To verify the user's signature $\sigma = (c, T, \pi)$ on message m, the verifier computes $e \leftarrow H_2(P_{\text{pub}}, ID, m, T, c)$ and checks if

$$c \stackrel{?}{=} H_1(P_{\text{pub}}, ID, cP_{\text{pub}} + e^{-1}(T - \pi P)).$$

If the equation holds with equality, return 1; otherwise, return 0. The verification works because of the following holds.

$$\begin{aligned} H_1(P_{\text{pub}}, ID, cP_{\text{pub}} + e^{-1}(T - \pi P)) &= H_1(P_{\text{pub}}, ID, cP_{\text{pub}} + e^{-1}(t - \pi)P)) \\ &= H_1(P_{\text{pub}}, ID, cP_{\text{pub}} + sP) \\ &= c. \end{aligned}$$

[2] For formality, one can include this set of parameters into the master public/private key pair (mpk, msk), and have this set of parameters be generated by some prime order elliptic curve cyclic subgroup generator.

4.1. Discussion

Note that T should be a nonce, that is, each value of T should only be used once. For each new signature generation, a new T should be used. In addition, the discrete logarithm of T should only be known to the signer. Otherwise, the user's secret key usk would be compromised. Hence in practice, the value of t should be destroyed once π is computed.

It is not difficult to see that the user-key generation algorithm is essentially the Schnorr signature [25] on the user's identity. One can also consider our entire IBS scheme to be a non-interactive proof system of a signature on the user's identity. The construction approach here is to have the trusted KGS generate a user's secret key as a signature on the user's identity, and then have the user conduct a non-interactive proof system of his secret key (i.e. the signature on his identity) using the transformation technique due to Fiat and Shamir [16] for converting the proof system to a signature scheme. This notion has first been discussed by Kurosawa and Heng [22] and Bellare et al. [5, Sec. 7 of the full paper].

A variant of BNN-IBS. We consider our IBS scheme as described above a variant of an IBS scheme called the BNN-IBS, which was proposed by Bellare et al. in [5, Sec. 7.3 of the full paper]. In the following, we explain the differences between them and show that our IBS scheme is more 'friendly' to low-power devices, as it requires less storage or computation resources.

In the BNN-IBS scheme, the component c in the user's secret key, usk, is replaced with R, and the computation of s is changed to $r + cx \mod p$. In other words, component c is computed as $H_1(P_{\text{pub}}, ID, sP - cP_{\text{pub}})$ in the BNN-IBS scheme, and the user's secret key usk now becomes $(R, s, ID, P_{\text{pub}})$ where $R = sP - cP_{\text{pub}}$. To sign a message m, the following steps are carried out, and a signature $\sigma = (R, S, T, \pi)$ is generated.

1. Compute $S = sP$, randomly pick $t \in_R \mathbb{Z}_p$ and compute $T = tP$.
2. Compute $e \leftarrow H_2(P_{\text{pub}}, ID, m, R, S, T)$ and $\pi = t + es \mod p$.

To verify the signature, $c = H_1(P_{\text{pub}}, ID, R)$ and $e = H_2(P_{\text{pub}}, ID, m, R, S, T)$ are first computed, and the following equations are then checked for equality.

$$\pi P \stackrel{?}{=} T + eS \tag{1}$$

$$S \stackrel{?}{=} R + cP_{\text{pub}}. \tag{2}$$

First of all, compared with our scheme, the signature size of the BNN-IBS scheme is larger. Secondly, the BNN-IBS scheme requires one more scalar multiplication for computing S in signature generation than ours. Although this additional operation can be saved by precomputing S and then caching it at the signer's side, it will then require the signer to have more memory space for caching this precomputed value. It turns out that the BNN-IBS scheme either requires one more scalar multiplication during the runtime of signature generation, or needs more storage space than our scheme.

The BNN-IBS scheme has been shown to be euf-cma-ida secure (Definition 1) in the random oracle model [6] under the assumption that the discrete logarithm problem is hard (in [5, Sec. 7.3 of the full paper]). In the following, we show that our IBS scheme is also euf-cma-ida secure.

Theorem 1. *If there exists a PPT adversary $\mathcal{A}$ which wins the game of Definition 1 for the IBS scheme proposed above with probability at least ϵ, then there exists a PPT adversary $\mathcal{B}$ which wins the game of Definition 1 for the BNN-IBS scheme with probability at least ϵ.*

Proof. We describe how to construct $\mathcal{B}$ when $\mathcal{A}$ is given. As defined in Definition 1, a challenger $\mathcal{S}$ simulates a game which captures the notions of adaptive chosen message attacks and ID attacks. At the end of the game, the adversary in the game is to output a triple (ID^*, m^*, σ^*) such that ID^* is not corrupted, m is not signed by ID^*, and σ^* is a valid signature of ID^* on message m^*.

Given an adversary $\mathcal{A}$ which breaks the euf-cma-ida security of the IBS scheme proposed above, we construct an adversary $\mathcal{B}$, which will break the euf-cma-ida security of the BNN-IBS scheme by running $\mathcal{A}$ and answering $\mathcal{A}$'s queries as follows.

- CreateUser: $\mathcal{B}$ relays such a query directly to $\mathcal{S}$ and relays back the answer from $\mathcal{S}$ to $\mathcal{A}$.
- Corrupt: $\mathcal{B}$ relays such a query to $\mathcal{S}$. Suppose the user's secret key usk returned by $\mathcal{S}$ is $(R, s, ID, P_{\text{pub}})$, $\mathcal{B}$ then queries $\mathcal{S}$ for $H_1(P_{\text{pub}}, ID, R)$. Suppose the answer of $\mathcal{S}$ is $\bar{c}$. $\mathcal{B}$ then sets c to $-\bar{c} \mod p$ (will be explained shortly) and sends $(c, s, ID, P_{\text{pub}})$ to $\mathcal{A}$ as the simulated answer to $\mathcal{A}$'s Corrupt query.

- Sign: $\mathcal{B}$ relays such a query to $\mathcal{S}$. Suppose $\mathcal{S}$'s answer to query $\mathsf{Sign}(ID, m)$ is $\sigma = (R, S, T, \pi)$, $\mathcal{B}$ then queries $\mathcal{S}$ for $H_1(P_{\text{pub}}, ID, R)$. Suppose the answer of $\mathcal{S}$ is $\bar{c}$, then it must be the case that $R = S - \bar{c}P_{\text{pub}}$ for having σ be valid according to (2) on Section 4.1. $\mathcal{B}$ sets c to $-\bar{c} \bmod p$ and sends $\sigma' = (c, T, \pi)$ to $\mathcal{A}$ as the simulated answer to $\mathcal{A}$'s $\mathsf{Sign}(ID, m)$ query.
- H_1: For any query of H_1 from $\mathcal{A}$, $\mathcal{B}$ relays it to $\mathcal{S}$. Suppose the answer of $\mathcal{S}$ is $\bar{c}$, $\mathcal{B}$ sets the answer for $\mathcal{A}$ to $-\bar{c} \bmod p$.
- H_2: For any query of H_2 from $\mathcal{A}$, $\mathcal{B}$ handles it in the following two cases depending on the query input.

  Case 1: If the query is on $(P_{\text{pub}}, \tilde{ID}, \tilde{m}, \tilde{T}, \tilde{c})$ where $\tilde{T}$ is some point and $-\tilde{c} \bmod p$ is the answer of $\mathcal{S}$ on query $H_1(P_{\text{pub}}, \tilde{ID}, \tilde{R})$ for some point $\tilde{R}$, then $\mathcal{B}$ queries $\mathcal{S}$ for $H_2(P_{\text{pub}}, \tilde{ID}, \tilde{m}, \tilde{R}, \tilde{S}, \tilde{T})$ where $\tilde{S} = \tilde{R} - \tilde{c}P_{\text{pub}}$. Suppose the answer of $\mathcal{S}$ is $\bar{e}$. $\mathcal{B}$ sets the answer for $\mathcal{A}$ to $-\bar{e} \bmod p$.

  Case 2: Otherwise (that is, at least one component of the input does not satisfy the form shown in Case 1), $\mathcal{B}$ randomly picks a value in $\mathbb{Z}_p$ as the answer to $\mathcal{A}$. Consistency (for replying the same value for the same queries) is maintained by having a table of queries, values, and answers maintained by $\mathcal{B}$.

When $\mathcal{A}$ outputs a valid signature (c^*, T^*, π^*) with message m^* and identity ID^*, due to the random oracle assumption, $\mathcal{A}$ must have queried for $c^* = H_1(P_{\text{pub}}, ID^*, R^*)$ where R^* is some point in order to pass all the steps of signature verification described in Section 4 (Note that the return of $\mathcal{S}$ for that H_1 query is $-c^* \bmod p$.). $\mathcal{B}$ sets $\sigma^* = (R^*, S^*, T^*, \pi^*)$ where $S^* = R^* - c^*P_{\text{pub}}$ and outputs the triple (ID^*, m^*, σ^*).

Analysis. To check the correctness of the simulation, first note that the $usk^{\text{New}} = (c, s, ID, P_{\text{pub}})$ of our protocol satisfies

$$c = H_1^{\text{New}}(P_{\text{pub}}, ID, sP + cP_{\text{pub}})$$

while the $usk^{\text{BNN-IBS}} = (\bar{c}, s, ID, P_{\text{pub}})$ of the BNN-IBS satisfies

$$\bar{c} = H_1^{\text{BNN-IBS}}(P_{\text{pub}}, ID, sP - \bar{c}P_{\text{pub}}).$$

By setting $c = -\bar{c} \bmod p$, we can see that the output of H_1^{New} is the complement of that of $H_1^{\text{BNN-IBS}}$. Hence, in the simulation above, we set the answer to $\mathcal{A}$ for queries of H_1 to the complement of the answer made by $\mathcal{S}$ for obtaining the reduction. Due to a similar reason, we also need to set the answer to $\mathcal{A}$ for queries of H_2 to the complement of the corresponding answer given by $\mathcal{S}$.

Obviously, the running time of $\mathcal{B}$ is a polynomial of that of $\mathcal{A}$. In addition, from $\mathcal{A}$'s point of view, all queries are simulated or relayed correctly. $\mathcal{A}$ cannot distinguish a simulated environment and a real game. Hence, if $\mathcal{A}$ makes a forgery of the IBS scheme proposed above, the reduction above correctly transforms the forgery to a forgery of the BNN-IBS scheme. □

5. An ID-based key exchange protocol

We now propose an ID-based key exchange (KE) protocol. The KE protocol is built using the IBS scheme described above. In the following, we first describe our scheme, then analyze its security under the Canetti–Krawczyk model [14] (in short, CK-model), and finally in the next section, we show that the scheme is much faster than the previously proposed protocols.

Let $k \in \mathbb{N}$ be a security parameter. Let A and B be the initiator and responder, respectively. They are identified by $ID_A, ID_B \in \{0, 1\}^*$, respectively. Let the secret key usk_A of A be $(c_A, s_A, ID_A, P_{\text{pub}})$ and the secret key usk_B of B be $(c_B, s_B, ID_B, P_{\text{pub}})$, which are generated according to the User-key Generation algorithm described in Section 4. Suppose A and B have a unique session-id ψ, shared already. We will give more details on the generation of ψ shortly. Below is the description of the ID-based KE protocol which is illustrated in Fig. 1.

Step 1. A picks $t_\alpha \in_R \mathbb{Z}_p$, computes $\alpha = t_\alpha P$, and sends (ID_A, ψ, α) to B.

Step 2. Upon receipt of (ID_A, ψ, α), B picks $t_\beta \in_R \mathbb{Z}_p$, computes $\beta = t_\beta P$ and sends $(ID_B, \psi, \beta, c_B, T_B, \pi_B)$ to A, where (c_B, T_B, π_B) is B's signature on message $m_B = (\psi, \beta, \alpha, ID_A)$. B also computes the session key $\gamma = t_\beta \alpha$ and erases t_β.

$$\begin{array}{lcl} A \rightarrow B & : & ID_A,\ \psi,\ \alpha \\ A \leftarrow B & : & ID_B,\ \psi,\ \beta,\ c_B, T_B, \pi_B \\ A \rightarrow B & : & ID_A,\ \psi,\ c_A, T_A, \pi_A \end{array}$$

Fig. 1. An ID-based KE protocol.

Step 3. Upon receipt of $(ID_B, \psi, \beta, c_B, T_B, \pi_B)$, A checks the correctness of each component in the incoming message and checks if the signature verification algorithm $Ver(P_{\text{pub}}, ID_B, m_B, (c_B, T_B, \pi_B))$ returns 1, where $m_B = (\psi, \beta, \alpha, ID_A)$. If the verification succeeds, A sends $(ID_A, \psi, c_A, T_A, \pi_A)$ to B where (c_A, T_A, π_A) is A's signature on $m_A = (\psi, \alpha, \beta, ID_B)$. A then computes $\gamma' = t_\alpha \beta$, erases t_α, and outputs the session key γ' under session-id ψ.

Step 4. Upon receipt of $(ID_A, \psi, c_A, T_A, \pi_A)$, B checks the correctness of each component in the incoming message and determines if the signature verification $Ver(P_{\text{pub}}, ID_A, m_A, (c_A, T_A, \pi_A))$ returns 1, where $m_A = (\psi, \alpha, \beta, ID_B)$. If the verification succeeds, B outputs the session key γ under session-id ψ.

As suggested by the authors of [14], in practice, the session-id ψ can be a pair (ψ_1, ψ_2), where ψ_1 is a value chosen by A such that (with very high probability) it is different from the values in other sessions of A and ψ_2 is chosen by B in a similar way. These values can be exchanged by the parties as a prologue [21]. Alternatively, ψ_1 can be included by A in the first message of the protocol above, and ψ_2 be included by B in the second message.

5.1. Security analysis

The protocol can be viewed as a Diffie–Hellman key exchange followed by a signature-based mutual authentication. T_A and T_B correspond to the key contributions of A and B, respectively, and signatures (c_A, T_A, π_A) and (c_B, T_B, π_B) of the IBS scheme proposed in the previous section correspond to the mutual authentication. In the following, we show that the protocol can be constructed using the modular approach introduced in [4,14].

The modular approach has two steps. In step one, a KE protocol is designed and shown to be secure in an ideal model called AM (Authenticated-links Model). In step two, the KE protocol is then *emulated* using some *Authenticator* such that the emulated protocol will be secure in a real-world model called UM (Unauthenticated-links Model). In AM, there is an adversary $\mathcal{A}$ sitting on the links between all the communicating parties, and is responsible for delivering messages faithfully. $\mathcal{A}$ cannot inject or modify messages. However, $\mathcal{A}$ can choose not to deliver it. There are also several oracles available for $\mathcal{A}$ to query, that include corrupt, reveal session state, reveal session output, etc. These oracles are used to capture the capabilities of $\mathcal{A}$ (for details, please refer to [14]). In UM, the adversary denoted by $\mathcal{U}$ is essentially the same as the AM adversary $\mathcal{A}$, but without the above restrictions on delivering messages. *Remark:* Each step in the above protocol description is considered to be atomic, and no corruption or reveal session can interrupt any step.

An *Authenticator* $\mathcal{C}$ is an algorithm that, on input a protocol τ, outputs another protocol $\mathcal{C}(\tau)$ such that $\mathcal{C}(\tau)$ *emulates* τ in UM. We say that $\mathcal{C}(\tau)$ *emulates* τ in UM if we can show that for any attack that the adversary $\mathcal{U}$ can launch against $\mathcal{C}(\tau)$ in UM, the AM adversary $\mathcal{A}$ can already launch the same attack against τ in AM. Please refer to [14] for the formal definition of *Authenticator*. The method of constructing an authenticator is given in [14], where a layered approach is used. In this approach, an authenticator $\mathcal{C}$ is constructed from **MT-authenticators**, and each of them emulates the basic message transmission protocol. The basic idea is that whenever a party A wants to send or receive a message, we emulate it using an MT-authenticator. Below is a signature-based MT-authenticator [4]. Suppose party A wants to send message m to party B. The following three-move protocol is carried out.

$$\begin{array}{lcl} A \rightarrow B & : & m \\ A \leftarrow B & : & m,\ N \\ A \rightarrow B & : & m,\ SIG_A(m, N, ID_B) \end{array}$$

$N \in_R \{0,1\}^k$ is a random challenge and SIG_A is the signature generation function of A. The signature scheme is required to be existential and unforgeable against the chosen message attack [19]. In the following, we show that the ID-based KE protocol shown in Fig. 1 can be constructed using this modular approach.

We start from the Protocol 2DH (a two-move Diffie–Hellman protocol in the AM) described in [14]. The protocol is reviewed as follows.

Table 1
Performance comparison

	Smart [27]	Yi [28]	Chen–Kudla [15]	Our protocol
Pairing	2	1	1	0
Scalar multiplication	2	3	4	6
Map-to-point hash	1	1	1	0

$$\begin{aligned} A \to B &: m_1 = (ID_A,\ \psi,\ \alpha = t_\alpha P) \\ A \leftarrow B &: m_2 = (ID_B,\ \psi,\ \beta = t_\beta P). \end{aligned}$$

In [14, Theorem 8], Protocol 2DH is shown to be secure in AM under the assumption that the Decisional Diffie–Hellman problem is hard. We then transform Protocol 2DH in AM to a protocol in UM by applying the signature-based MT-authenticator above to m_1 and m_2 of Protocol 2DH above. After that, we use the optimization technique described in [4,14] to reduce the number of message flows from six to three. The resulting protocol is illustrated below[3].

$$\begin{aligned} A \to B &: ID_A,\ \psi,\ \alpha \\ A \leftarrow B &: ID_B,\ \psi,\ \beta,\ SIG_B(m_B) \\ A \to B &: ID_A,\ \psi,\ SIG_A(m_A). \end{aligned}$$

In the protocol, $m_B = (ID_B, \psi, \beta, \alpha, ID_A)$ and $m_A = (ID_A, \psi, \alpha, \beta, ID_B)$. To instantiate the signature scheme using the IBS scheme described in Section 4, we have SIG_A become (c_A, T_A, π_A) and SIG_B become (c_B, T_B, π_B). In addition, from the computation of e in the signature generation phase, we can see that some components in m_A and m_B are also redundant. By further eliminating those redundant components, the final optimized protocol will become the one shown in Fig. 1.

In the following, we further evaluate our ID-based KE protocol by considering some additional features and attacks that are not captured by the CK-model.

KGS forward secrecy. Our ID-based KE protocol constructed above using the modular approach of [4,14] is essentially the Protocol SIG-DH of [14] which is secure under the CK-model. This also implies that the protocol satisfies perfect forward secrecy. Although the CK-model does not capture KGS forward secrecy, we can still see that our protocol supports KGS forward secrecy, as session keys are solely derived from contributions α and β.

Key compromise impersonation resilience (KCIR). As defined in [12], a protocol provides resistance to key compromise impersonation if the compromising of the long-term secret of a party A does not allow the adversary to masquerade *to* A as a different party. To see that compromising A's secret usk_A does not allow the adversary to masquerade to A as B, we notice that the adversary has to provide a signature of B in the second message of the protocol before A accepts. As long as α is a nonce and B's signature is existentially unforgeable, the adversary cannot provide a correct signature. Similar reasons can be applied to explain the KCIR of B as well.

6. Performance analysis

In Table 1, we summarize the number of different operations of some well-known ID-based KE protocols and our protocol proposed above. We ignore the time taken by conventional hash operations and point addition operations as they are much more efficient when compared with pairings, scalar multiplications, and map-to-point hash operations.

According to the state-of-the-art results in [1,2], one pairing operation requires at least 10 times more multiplications in the underlying finite field than a point scalar multiplication does in the same finite field. Hence, those pairing-based KE protocols are much slower than the one proposed in this paper. When compared with old, non-pairing based protocols such as [24,18,17], our protocol is also much faster because each communicating party of these old, non-pairing based protocols needs to do expensive modular exponentiation operations. For protocols proposed in [20], they can be implemented under an elliptic curve group, and one of the protocols in [20] is also believed to support KGS forward secrecy. On its performance, the protocol requires each communicating party to

[3] This is identical to the Protocol SIG-DH in [14].

carry out seven scalar multiplications. Hence, it is slightly less efficient than our protocol. In addition, the protocol is not known to be provably secure. In [13], Burmester showed that the protocol is vulnerable to an attack called the triangle attack. Since the triangle attack is captured in the CK-model and our protocol is proven secure in this model, our protocol is not vulnerable to this attack.

7. Conclusion

In this paper, we proposed an ID-based signature scheme and showed that it is a variant of the BNN-IBS scheme proposed by Bellare et al. [5]. Our scheme is more efficient than the BNN-IBS one, as it requires either less storage space or less runtime computation. Using our ID-based signature scheme, we proposed a new ID-based KE protocol. The protocol does not require any pairing operation or map-to-point hash operation. It also supports the strongest KGS forward secrecy. For security analysis, we show that it can be constructed using the modular approach of [4]. As for its efficiency, we showed that it is faster than all comparable ID-based KE protocols.

For further reading

[30]

Acknowledgement

The third author was supported by a grant from CityU (Project No. 7001959).

References

[1] P. Barreto, H. Kim, B. Lynn, M. Scott, Efficient algorithms for pairing-based cryptosystems, in: Proc. CRYPTO 2002, in: LNCS, vol. 2442, Springer-Verlag, 2002, pp. 354–368.
[2] P. Barreto, B. Lynn, M. Scott, On the selection of pairing-friendly groups, in: Selected Areas in Cryptography, SAC 2003, in: LNCS, vol. 3006, Springer-Verlag, 2004, pp. 17–25.
[3] P.S.L.M. Barreto, B. Libert, N. McCullagh, J.-J. Quisquater, Efficient and provably-secure identity-based signatures and signcryption from bilinear maps, in: Proc. ASIACRYPT 2005, in: LNCS, vol. 3788, Springer-Verlag, 2005, pp. 515–532.
[4] M. Bellare, R. Canetti, H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols, in: Proc. 30th ACM Symp. on Theory of Computing, ACM, 1998, pp. 419–428.
[5] M. Bellare, C. Namprempre, G. Neven, Security proofs for identity-based identification and signature schemes, in: Proc. EUROCRYPT 2004, in: LNCS, vol. 3027, Springer-Verlag, 2004, pp. 268–286. (Full paper is available at Bellare's homepage URL: http://www-cse.ucsd.edu/users/mihir).
[6] M. Bellare, P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, in: 1st ACM Conference on Computer and Communications Security, ACM, Fairfax, 1993, pp. 62–73.
[7] M. Bellare, P. Rogaway, Entity authentication and key distribution, in: Proc. CRYPTO 93, in: LNCS, vol. 773, Springer, 1994, pp. 232–249.
[8] M. Bellare, P. Rogaway, Provably secure session key distribution — the three party case, in: Proc. 27th ACM Symp. on Theory of Computing, ACM, Las Vegas, 1995, pp. 57–66.
[9] D. Boneh, X. Boyen, Short signatures without random oracles, in: Proc. EUROCRYPT 2004, in: LNCS, vol. 3027, Springer-Verlag, 2004, pp. 56–73.
[10] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Proc. CRYPTO 2001, in: LNCS, vol. 2139, Springer-Verlag, 2001, pp. 213–229.
[11] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, in: Proc. ASIACRYPT 2001, in: LNCS, vol. 2248, Springer-Verlag, 2001, pp. 514–532.
[12] C. Boyd, A. Mathuria, Protocols for Authentication and Key Establishment, Springer-Verlag, 2003.
[13] M. Burmester, On the risk of opening distributed keys, in: Proc. CRYPTO 94, in: LNCS, vol. 839, Springer, 1994, pp. 308–317.
[14] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: Proc. EUROCRYPT 2001, in: LNCS, vol. 2045, Springer-Verlag, 2001, pp. 453–474. http://eprint.iacr.org/2001/040/.
[15] L. Chen, C. Kudla, Identity based authenticated key agreement protocols from pairings, Cryptology ePrint Archive, Report 2002/184, 2002. http://eprint.iacr.org/.
[16] A. Fiat, A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, in: Proc. CRYPTO 86, in: LNCS, vol. 263, Springer-Verlag, 1987, pp. 186–199.
[17] M. Girault, Self-certified public keys, in: Proc. EUROCRYPT 91, in: LNCS, vol. 547, Springer-Verlag, 1991, pp. 490–497.
[18] M. Girault, J.-C. Paillès, An identity-based scheme providing zero-knowledge authentication and authenticated key exchange, in: European Symposium on Research in Computer Security, AFCET, October 1990, pp. 173–184.
[19] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen-message attack, SIAM J. Comput. 17 (2) (1988) 281–308.

[20] C. Günther, An identity-based key exchange protocol, in: Proc. EUROCRYPT 89, in: LNCS, vol. 434, Springer-Verlag, 2000, pp. 29–37.
[21] D. Harkins, C. Kaufman, R. Perlman, The internet key exchange (IKE) protocol <draft-ietf-ipsec-ikev2-00.txt>. INTERNET-DRAFT, November 2001.
[22] K. Kurosawa, S. Heng, From digital signature to ID-based identification/signature, in: Proc. of PKC 2004, in: LNCS, vol. 2947, Springer-Verlag, 2004, pp. 248–261.
[23] N. McCullagh, P. Barreto, A new two-party identity-based authenticated key agreement, in: CT-RSA 2005, in: LNCS, vol. 3376, Springer-Verlag, 2005, pp. 262–274.
[24] E. Okamoto, Key distribution systems based on identification information, in: Proc. CRYPTO 87, in: LNCS, vol. 293, Springer-Verlag, 1988, pp. 194–202.
[25] C. Schnorr, Efficient identification and signatures for smart cards, in: G. Brassard (Ed.), Proc. CRYPTO 89, in: LNCS, vol. 435, Springer, 1990, pp. 239–252.
[26] A. Shamir, Identity-based cryptosystems and signature schemes, in: Proc. CRYPTO 84, in: LNCS, vol. 196, Springer, 1984, pp. 47–53.
[27] N. Smart, Identity-based authenticated key agreement protocol based on Weil pairing, IEE Electron. Lett. 38 (13) (2002) 630–632.
[28] X. Yi, Efficient ID-based key agreement from Weil pairing, IEE Electron. Lett. 39 (2) (2003) 206–208.
[29] F. Zhang, R. Safavi-Naini, W. Susilo, An efficient signature scheme from bilinear pairings and its applications, in: Proc. of PKC 2004, in: LNCS, vol. 2947, Springer-Verlag, 2004, pp. 277–290.
[30] R.W. Zhu, G. Yang, D.S. Wong, An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices, in: Internet and Network Economics: First International Workshop, WINE 2005, in: LNCS, vol. 3828, Springer-Verlag, 2005, pp. 500–509.